A Tustin-Based Discrete-Time Implementation of the Generalized Clegg Integrator

Luke F. van Eijk
[1] *Department of Precision and Microsystems Engineering*
Delft University of Technology
Delft, The Netherlands
[2] *Center of Competency, ASMPT*
Beuningen, The Netherlands
L.F.vanEijk@tudelft.nl

Dragan Kostić
Center of Competency
ASMPT
Beuningen, The Netherlands
dragan.kostic@asmpt.com

S. Hassan HosseinNia
Department of Precision and Microsystems Engineering
Delft University of Technology
Delft, The Netherlands
S.H.HosseinNiaKani@tudelft.nl

2025 13th International Conference on Control, Mechatronics and Automation (ICCMA) | 979-8-3315-9141-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCMA67641.2025.11369644

***Abstract*—This paper proposes a novel discrete-time (DT) implementation of the generalized Clegg integrator (GCI), which is an integrator that resets its state to a fraction of the original state when its input is equal to zero. The implementation is derived by discretizing a continuous-time (CT) GCI using the Tustin discretization method. By means of a numerical validation it is shown that the state of the DT GCI is identical to its CT counterpart when both are subject to an input which is linearly interpolated between samples, as expected when using this discretization method. For a general CT input which is not linearly interpolated between samples, a numerical comparison is made between the state of the novel DT GCI and the CT GCI. At samples with linear behaviour, the state mismatch is equivalent to the one observed between their linear counterparts. At samples with resetting behaviour, the mismatch even reduces compared to previous samples, as a consequence of (partially) resetting the state mismatch.**

***Index Terms*—Generalized Clegg integrator, discrete-time, Tustin discretization, reset control, nonlinear control**

I. Introduction

Reset control is a nonlinear technique where controller states are reset when certain conditions are satisfied [1]. The most basic example of a reset controller is the Clegg integrator, which is an integrator that resets its state to zero when its input is equal to zero [2]. Reset controllers have gained interest because they can overcome certain inherent performance limitations of linear controllers. This has been theoretically proven for various overshoot performance limitations of linear control in, e.g., [3] and [4]. Furthermore, it has been experimentally demonstrated on various applications, such as in hard disk drives [5], semiconductor manufacturing equipment [6], heat exchangers [7], power converters [8], and in-line pH control [9].

Reset controllers where (a part of) the controller states are (partially) reset when their input is equal to zero, are recently gaining popularity in industry for the control of linear plants [6]. Namely, intuitive frequency-domain based design tools have been developed for this class of reset controllers, similar to existing design tools that are widely used in industry for the design of linear controllers [10]–[13]. An intuitive graphical stability check for closed-loop reset control systems has been developed in [10], which can be assessed based solely on a (measured) frequency-response function (FRF) of the plant. For many mechatronic applications, such FRFs are already available or can be identified with only a limited amount of measurement data [14]. Furthermore, open- and closed-loop steady-state performance of reset control systems can be analysed using, e.g., the tools developed in [11]–[13].

While reset controllers are in most mechatronic applications eventually implemented in discrete-time (DT), the available literature dominantly investigates continuous-time (CT) reset controllers. Several DT reset controller implementations have been suggested, such as the ones used in [6] and [15]. At samples with linear behaviour (no state resets), the DT implementation is straightforward: it is the same as implementing a linear controller in discrete-time. However, it is unclear how the DT implementations of the reset condition have been derived, as well as the state update laws at samples for which this condition is satisfied. Given that there is no systematic derivation, it is also unclear whether these DT implementations accurately emulate their CT counterpart.

The aim of this work is to take an initial step towards the systematic derivation of a DT reset controller implementation that accurately emulates its CT counterpart. To that end, we derive a novel DT implementation of a generalized Clegg integrator (GCI), which is a CI that resets its state to a fraction of its original state [16]. We specifically use the Tustin discretization method, because it is one of the most popular methods for controller discretization, especially when one desires a good frequency-domain match between the CT and DT implementations [17]. Note that other discretization methods can also be used to derive different DT GCI implementations, but this is outside of the scope of this work.

The remainder of this paper is organized as follows. In Section II we provide the necessary preliminaries of the Tustin discretization method and the CT GCI. Next, in Section III we derive a DT GCI implementation using the Tustin method. Subsequently, in Section IV we numerically validate the cor-

This work was supported by ASMPT.

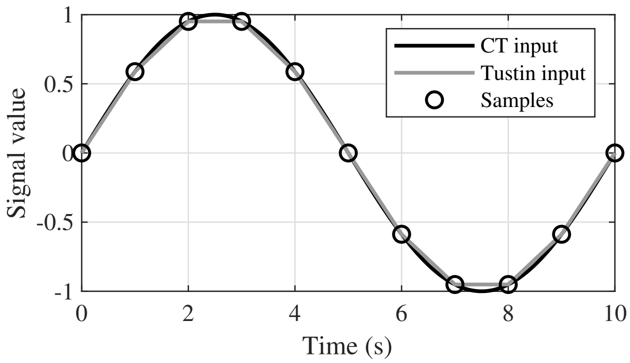


Fig. 1: Assumption on inter-sample behaviour using the Tustin method, for a CT input $e(t) = \sin(0.2\pi t)$ and a sample time $T_{\mathrm{s}} = 1$.

rectness of the proposed DT implementation, and compare its behaviour to that of a CT GCI. Finally, we provide conclusions and future work in Section V.

II. Preliminaries

A. Tustin-Based Digital Implementation of an Integrator

Consider a CT linear integrator, given in state-space representation by

$$\dot{x}(t) = e(t) \tag{1}$$

with state $x \in \mathbb{R}$, input $e \in \mathbb{R}$, and time $t \in \mathbb{R}$. The state of the integrator is then given by

$$x(t) = x(t^*) + \int_{t^*}^{t} e(\tau)\mathrm{d}\tau \quad \forall t \geq t^* \tag{2}$$

with $t^* \in \mathbb{R}$.

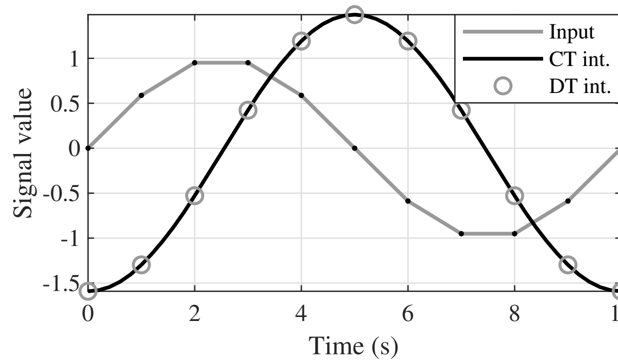
In digital systems, the input can only be sampled at discrete time-intervals. In this work we focus on digital systems with sampling at equidistant time-intervals $T_{\mathrm{s}} \in \mathbb{R}_{>0}$, where T_{s} is also known as the sampling time. In other words, the system is sampled at times $t_k := (k-1)T_{\mathrm{s}}$, with sample number $k \in \mathbb{N}$. To emulate the behaviour of the CT integrator by means of a digitally implemented version, an assumption has to be made on the behaviour of the input in between samples. In this work we focus on the trapezoidal rule, which is also known as the Tustin method [17]. This method assumes that the input is linearly interpolated between samples, such that the input is approximated by

$$e(t) = e(t_k) + [e(t_{k+1}) - e(t_k)] \frac{t - t_k}{T_{\mathrm{s}}} \;\; \forall t \in [t_k, t_{k+1}). \tag{3}$$

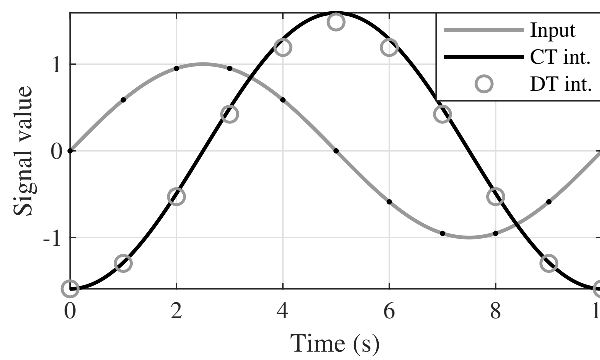
As an example, the resulting approximation is shown in Fig. 1 for a sinusoidal input.

If we assume that the input is linearly interpolated between samples, we can explicitly derive the integrator state at samples k by using (2), which leads to the DT integrator update law given by

$$x_k = x_{k-1} + \frac{T_{\mathrm{s}}(e_{k-1} + e_k)}{2}. \tag{4}$$



(a) Linearly interpolated input.



(b) Real CT input.

Fig. 2: Comparison of integrator states obtained with a sinusoidal input $e(t) = \sin(\omega t)$, $\omega = 0.2\pi$, for the CT integrator (1) with initial condition $x(0) = -\frac{1}{\omega}$ and for the DT integrator (4) with sample time $T_{\mathrm{s}} = 1$ and initial condition $x_1 = -\frac{1}{\omega}$.

In Fig. 2a we compare the integrator states of the CT integrator and DT integrator when both are subject to a sinusoidal input which is linearly interpolated between the samples of the DT GCI. As expected, the states of both elements are identical at all samples. In Fig. 2b we compare the integrator states of both elements when they are subject to the real (non-interpolated) CT sinusoidal input. Note that the DT integrator state values are identical to the ones in Fig. 2a, because the input is equivalent at all samples. However, the CT integrator state values have changed, which results in a mismatch with the DT integrator state values.

B. Continuous-Time Generalized Clegg Integrator

Consider the GCI, which is an integrator that resets its state to a fraction of the original state when the input is equal to zero [16]. The CT GCI is given in state-space representation by

$$R_{\mathrm{CT}} := \begin{cases} \dot{x}(t) = e(t) & \text{if } e(t) \neq 0 \\ x(t^+) = \gamma x(t) & \text{if } e(t) = 0 \end{cases} \tag{5}$$

with reset fraction $\gamma \in (-1, 1)$ and $t^+ := \lim_{\tau \to t, \tau > t} \tau$. As an example, the response of a GCI to a sinusoidal input is visualized in Fig. 3.

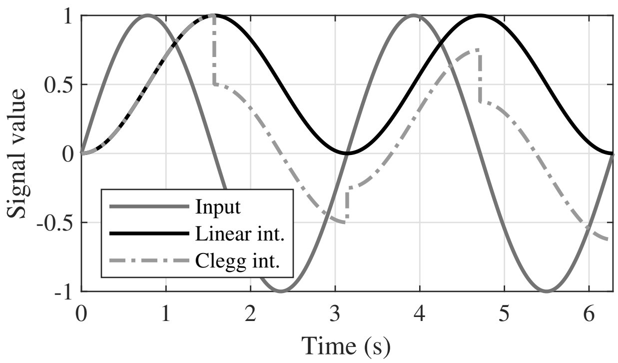


Fig. 3: Comparison of CT linear integrator state and CT GCI state ($\gamma = 0.5$) for an input $e(t) = \sin(2t)$ and initial condition $x(0) = 0$.

III. Tustin-Based Discretization of Generalized Clegg Integrator

In this section we develop the DT implementation of the GCI described in Section II-B, using the Tustin method described in Section II-A. First, in Section III-A we derive a DT reset instant detection algorithm. Second, in Section III-B we derive the DT update law for the case where a zero-crossing is detected. Finally, in Section III-C we derive the complete DT update law of the GCI.

A. Discrete-Time Reset Instant Detection

As described in (5), a GCI resets its state when the input is equal to zero. A bold method to discretize the reset condition would be to check, at sample k, whether $e_k = 0$. However, it is also possible that the input is equal to zero in between samples, which then is not detected by the algorithm. Therefore, at sample k we need to check whether the input is equal to zero for any time $t \in (t_{k-1}, t_k]$ when we linearly interpolate between the input at the current and previous sample. To check this, we make use of Fig. 4, which illustrates the different situations that can occur. In specific, we differentiate between the cases where we linearly interpolate between inputs that are either positive, zero, and/or negative. This leads to a total of nine possible situations that can occur. The first four situations, for which the input is never equal to zero, are given by

(S_1) $e_{k-1} > 0$ and $e_k > 0$,
(S_2) $e_{k-1} < 0$ and $e_k < 0$,
(S_3) $e_{k-1} = 0$ and $e_k > 0$,
(S_4) $e_{k-1} = 0$ and $e_k < 0$.

In situations (S_1) and (S_2) we either draw a straight line between two positive values or between two negative values, resulting in a line that is nonzero for all time $t \in (t_{k-1}, t_k]$. For situations (S_3) and (S_4), the input is equal to zero at time t_{k-1}. However, this is not within the time-interval we are interested in. The next four situations, for which the input is once equal to zero during the time-interval $t \in (t_{k-1}, t_k]$, are given by

(S_5) $e_{k-1} > 0$ and $e_k < 0$,
(S_6) $e_{k-1} < 0$ and $e_k > 0$,

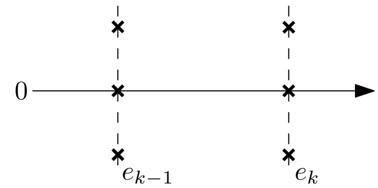


Fig. 4: Schematic illustration of inputs e_{k-1} and e_k, where we differentiate between values that are positive, zero, or negative.

(S_7) $e_{k-1} > 0$ and $e_k = 0$,
(S_8) $e_{k-1} < 0$ and $e_k = 0$.

In situations (S_5) and (S_6), the straight line between the two inputs crosses zero between samples. Essentially, this means that a sign-switch occurs, which is the case if and only if $e_k e_{k-1} < 0$. In situations (S_7) and (S_8), the straight line is exactly equal to zero at time t_k. Therefore, we can detect situations (S_5)-(S_8) by checking whether the input is in the set

$$J_k^1 = \{(e_k, e_{k-1}) \in \mathbb{R}^2 \mid e_k e_{k-1} < 0 \vee ... \\ (e_k = 0 \wedge e_{k-1} \neq 0)\}. \quad (6)$$

Finally, the ninth situation, for which the input is equal to zero for all times $t \in (t_{k-1}, t_k]$, is given by

(S_9) $e_{k-1} = 0$ and $e_k = 0$.

We can detect this last situation by checking whether the input is in the set

$$J_k^\infty = \{(e_k, e_{k-1}) \in \mathbb{R}^2 \mid e_k = 0 \wedge e_{k-1} = 0\}. \quad (7)$$

For later use in Section III-C, we also define the set

$$F_k = \{(e_k, e_{k-1}) \in \mathbb{R}^2 \mid (e_k, e_{k-1}) \notin (J_k^1 \cup J_k^\infty)\}, \quad (8)$$

which corresponds to situations (S_1)-(S_4).

B. Update Law With One Reset Instant

We now derive the DT GCI update law in the case that the input is within the set J_k^1 as in (6). In other words, the input crosses zero at one point during the sample, such that there is one reset action. To derive the update law, we use Fig. 5, which schematically illustrates the inter-sample behaviour in the case of a zero-crossing. According to the Tustin method, we assume that between samples the input is linearly interpolated between the sampled inputs. Based on that assumption, we find that the input is equal to zero at time

$$t_{\mathrm{cr}} = t_{k-1} + T_{\mathrm{s}} \cdot \frac{-e_{k-1}}{e_k - e_{k-1}} \quad (9)$$

or, when alternatively written, at time

$$t_{\mathrm{cr}} = t_k - T_{\mathrm{s}} \cdot \frac{e_k}{e_k - e_{k-1}}. \quad (10)$$

Before the hypothetical zero-crossing time t_{cr}, the integrator still integrates the input over the time-interval ranging from time t_{k-1} until time t_{cr}. The duration of this interval is

$$T_1 = t_{\mathrm{cr}} - t_{k-1}, \quad (11)$$

which, after substituting (9), results in

$$T_1 = \frac{-T_\mathrm{s} e_{k-1}}{e_k - e_{k-1}}. \tag{12}$$

As illustrated in Fig. 5, the integrated area in that time-interval is given by

$$A_1 = \frac{T_1 e_{k-1}}{2}, \tag{13}$$

which, by substituting (12), can be written as

$$A_1 = \frac{-T_\mathrm{s} e_{k-1}^2}{2(e_k - e_{k-1})}. \tag{14}$$

At time-instant t_cr, the integrator state then hypothetically resets to

$$x_\mathrm{cr} = \gamma(x_{k-1} + A_1), \tag{15}$$

$$x_\mathrm{cr} = \gamma \left(x_{k-1} - \frac{T_\mathrm{s} e_{k-1}^2}{2(e_k - e_{k-1})} \right). \tag{16}$$

Subsequently, integration continues over the time-interval ranging from time t_cr until time t_k. The duration of this interval is

$$T_2 = t_k - t_\mathrm{cr}, \tag{17}$$

which, after substituting (10), results in

$$T_2 = \frac{T_\mathrm{s} e_k}{e_k - e_{k-1}}. \tag{18}$$

As illustrated in Fig. 5, the integrated area in that time-interval is given by

$$A_2 = \frac{T_2 e_k}{2}, \tag{19}$$

which, after substituting (18), can be written as

$$A_2 = \frac{T_\mathrm{s} e_k^2}{2(e_k - e_{k-1})}. \tag{20}$$

Therefore, the integrator state after a zero-crossing detection is given by

$$x_k = x_\mathrm{cr} + A_2, \tag{21}$$

which, after substituting (16) and (20) into (21), and rewriting, yields

$$x_k = \gamma x_{k-1} + \frac{T_\mathrm{s}(e_k^2 - \gamma e_{k-1}^2)}{2(e_k - e_{k-1})}. \tag{22}$$

C. Complete Discrete-Time GCI Implementation

Finally, we derive the complete DT GCI implementation along the line of the Tustin method. If no zero-crossing is detected – the input is in set F_k as in (8) – the GCI should follow the update law of the linear DT Tustin integrator, as given in (4). If one zero-crossing is detected – the input is in set J_k^1 as in (6) – the GCI should follow the update law as derived in (22). Finally, if the input is equal to zero during the entire sample – the input is in set J_k^∞ as in (7) – there are an infinite amount of resets during the sample. For the CT GCI in (5) we only consider reset fractions $\gamma \in (-1, 1)$, meaning that the absolute value of the integrator state reduces with each reset. Therefore, when (hypothetically) resetting an

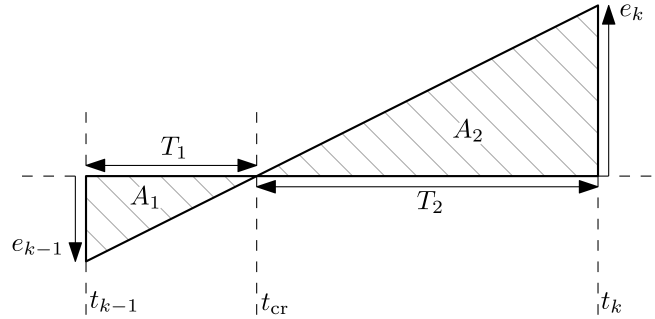


Fig. 5: Schematic illustration of zero-crossing time-instant t_cr and integration areas A_1 and A_2, used for derivation of the DT integrator update law when one zero-crossing is detected.

infinite amount of times, the integrator state becomes zero. Hence, the complete update law of the DT GCI is given by

$$R_\mathrm{DT} := \begin{cases} x_k = x_{k-1} + \frac{T_\mathrm{s}(e_{k-1}+e_k)}{2} & \\ & \text{if } (e_k, e_{k-1}) \in F_k \\ x_k = \gamma x_{k-1} + \frac{T_\mathrm{s}(e_k^2 - \gamma e_{k-1}^2)}{2(e_k - e_{k-1})} & \\ & \text{if } (e_k, e_{k-1}) \in J_k^1 \\ x_k = 0 & \text{if } (e_k, e_{k-1}) \in J_k^\infty \end{cases} \tag{23}$$

with sets J_k^1, J_k^∞, and F_k, as in (6)-(8), respectively.

Note that in order for the DT GCI's output to effectively emulate the output of its CT counterpart, it is important that the CT input signal can be reasonably well approximated by the linearly interpolated signal. For example, consider an input signal which contains an oscillation frequency above the sampling frequency. In such a case, the predicted zero-crossing time t_cr might not be accurate. Furthermore, the predicted amount of zero-crossings (i.e. reset instants) might not be correct. Therefore, it is important that the sampling frequency is chosen sufficiently large compared to the dominant vibration frequencies that are present in the input signal.

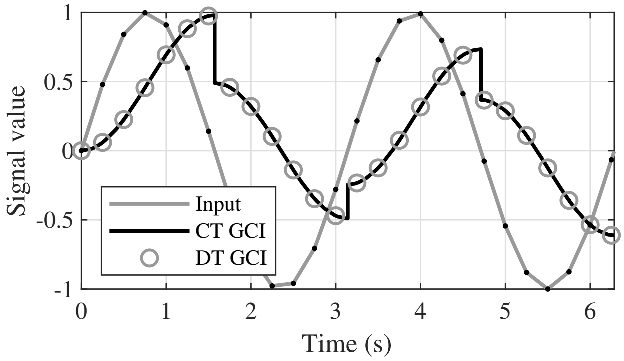
Remark 3.1: It is interesting to note that time-regularized versions of the CT GCI, which only allow a new reset after a certain time-period has passed, have been investigated before (see, e.g., [18]). Such an approach can for example be used to prevent excessive resetting due to high-frequent measurement noise. In situations (S_5)-(S_8) we are essentially introducing a similar form of time-regularization. Namely, even if high-frequent vibrations in the input signal cause multiple zero-crossings in between a sample, only one reset is triggered.

IV. Numerical Validation

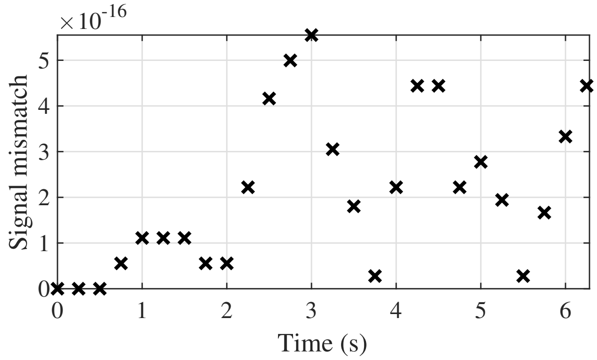
In this section we numerically validate the developed DT GCI implementation. First, in Section IV-A we numerically verify whether the update law has been derived correctly. Second, in Section IV-B we illustrate the effectiveness of the DT GCI in emulating the response of the CT GCI.

A. Validation of Update Law Derivation

To verify whether the update law in (23) has been derived correctly, we compare the behaviour of the CT GCI and DT

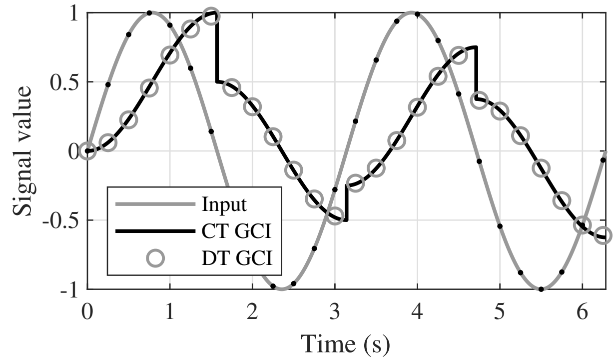


(a) Comparison of CT and DT GCI states.

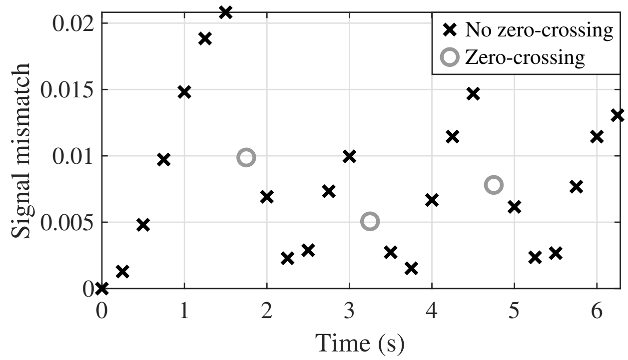


(b) Absolute mismatch between CT and DT GCI states.

Fig. 6: Response comparison of the CT GCI (5) with initial condition $x(0) = 0$, and the DT GCI (23) with $T_\mathrm{s} = 0.25$ and initial condition $x_1 = 0$, both with $\gamma = 0.5$, for a sinusoidal input $e(t_k) = \sin(2t_k)$ that is linearly interpolated between samples.



(a) Comparison of CT and DT GCI states.



(b) Absolute mismatch between CT and DT GCI states.

Fig. 7: Response comparison of the CT GCI (5) with initial condition $x(0) = 0$, and the DT GCI (23) with $T_\mathrm{s} = 0.25$ and initial condition $x_1 = 0$, both with $\gamma = 0.5$, for a sinusoidal input $e(t) = \sin(2t)$.

GCI when both are subject to a sinusoidal input which is linearly interpolated between the samples of the DT GCI, as illustrated in Fig. 6. The states of both elements, in response to the linearly interpolated sinusoidal input, are shown in Fig. 6a. Again, we want to stress that the CT GCI is also subject to a sinusoidal input which is linearly interpolated with the same sample time as for the DT GCI. In theory, in that case the states of the CT and DT GCI should be identical at all samples. Therefore, we visualize the absolute mismatch between the states of the DT and CT GCI at each sample in Fig. 6b. The mismatch is of the order 10^{-16}, which is of a similar level as the precision of the numerical solver that has been used [19]. Therefore, we can conclude that the developed DT GCI update law in (23) is correctly derived.

B. Comparison of CT and DT GCI

To illustrate up to which extent the developed DT GCI implementation can emulate the state of the CT GCI, we compare the behaviour of both elements in Fig. 7. The states of both elements, in response to a CT (non-interpolated) sinusoidal input, are shown in Fig. 7a. Furthermore, the absolute mismatch between the CT and DT states is displayed in Fig. 7b. One can observe in the figure that there is no zero-crossing of the input in the first seven samples. Therefore, the mismatch between the states of the CT and DT GCI is equivalent to that of a linear integrator, as earlier illustrated in Fig. 2b. If we focus specifically on the samples at which a zero-crossing occurs, i.e., a state reset happens, the mismatch even consistently reduces. This illustrates the effectiveness of the developed update law in the situation that a zero-crossing is detected. When we look in more detail at Fig. 7b, we can observe that the mismatch reduces with roughly a factor γ at samples where a state reset occurs. We believe this is because the mismatch built up during the linear behaviour is partially taken away due to the reset action.

V. Conclusion

In this work we proposed a formal discrete-time (DT) implementation of a generalized Clegg integrator (GCI), using the Tustin discretization method. We showed numerically that the states of the continuous-time (CT) GCI and novel DT GCI are identical when both are subject to an input which is linearly interpolated between samples. This illustrates that the DT implementation has been derived correctly. Furthermore, we numerically studied the difference between the DT GCI and CT GCI, when both elements are subject to the same CT (non-interpolated) input. At samples where the GCI behaves as a linear integrator, the mismatch is equivalent to the one observed with a Tustin-discretized linear integrator. At samples where resetting behaviour occurs, the mismatch between the DT and CT GCI even drops by a fraction which is roughly equal to the GCI's reset fraction.

As part of future work, we will experimentally validate the effectiveness of the developed DT GCI implementation. Furthermore, we will use the Tustin discretization method to derive a DT implementation of a generalized first-order reset element [16] and higher-order reset elements [20].

REFERENCES

[1] A. Baños and A. Barreiro, *Reset Control Systems*, 1st ed. Springer London, 2011.

[2] J. C. Clegg, "A nonlinear integrator for servomechanisms," *Trans. Amer. Inst. Electr. Eng. II, Appl. Ind.*, vol. 77, no. 1, pp. 41–42, 1958.

[3] O. Beker, C. Hollot, and Y. Chait, "Plant with integrator: an example of reset control overcoming limitations of linear feedback," *IEEE Trans. Autom. Control*, vol. 46, no. 11, pp. 1797–1799, 2001.

[4] G. Zhao, D. Nešić, Y. Tan, and C. Hua, "Overcoming overshoot performance limitations of linear systems with reset control," *Automatica*, vol. 101, pp. 27–35, 2019.

[5] Y. Li, G. Guo, and Y. Wang, "Reset control for midfrequency narrowband disturbance rejection with an application in hard disk drives," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1339–1348, 2011.

[6] D. Caporale, L. F. van Eijk, N. Karbasizadeh, S. Beer, D. Kostić, and S. H. HosseinNia, "Practical implementation of a reset controller to improve performance of an industrial motion stage," *IEEE Trans. Control Syst. Technol.*, vol. 32, no. 4, pp. 1451–1462, 2024.

[7] A. Vidal and A. Baños, "Reset compensation for temperature control: Experimental application on heat exchangers," *Chem. Eng. J.*, vol. 159, no. 1, pp. 170–181, 2010.

[8] U. Raveendran Nair, R. Costa-Castelló, and A. Baños, "Reset control for dc–dc converters: An experimental application," *IEEE Access*, vol. 7, pp. 128 487–128 497, 2019.

[9] J. Carrasco and A. Baños, "Reset control of an industrial in-line ph process," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 4, pp. 1100–1106, 2012.

[10] A. A. Dastjerdi, A. Astolfi, and S. H. HosseinNia, "Frequency-domain stability methods for reset control systems," *Automatica*, vol. 148, p. 110737, 2023.

[11] L. F. van Eijk, D. Kostić, M. Khosravi, and S. H. HosseinNia, "Higher order sinusoidal-input describing function analysis for a class of multiple-input multiple-output convergent systems," *IEEE Trans. Autom. Control*, vol. 70, no. 1, pp. 673–680, 2025.

[12] L. F. van Eijk, D. Kostić, and S. Hassan HosseinNia, "Frequency response analysis of general zero-crossing reset control systems," *IEEE Contr. Syst. Lett.*, vol. 9, pp. 1105–1110, 2025.

[13] X. Zhang and S. H. HosseinNia, "Higher-order sinusoidal input describing functions for open-loop and closed-loop reset control with application to mechatronics systems," *Mech. Syst. Signal Process.*, vol. 236, p. 112881, 2025.

[14] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, 2nd ed. Wiley-IEEE Press, 2012.

[15] V. Toochinda and A. Suwannakom, "Practical discrete-time implementation of reset control systems," in *Proc. Int. Conf. Control and Robot. Eng.*, 2017, pp. 86–91.

[16] Y. Guo, Y. Wang, and L. Xie, "Frequency-domain properties of reset systems with application in hard-disk-drive systems," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 6, pp. 1446–1453, 2009.

[17] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Ellis-Kagle Press, 1997.

[18] B. Kieft, S. H. HosseinNia, and N. Saikumar, "Time regularization as a solution to mitigate quantization induced performance degradation," in *Proc. Eur. Control Conf.*, 2021, pp. 2458–2464.

[19] The MathWorks Inc., "Simulink," August 2025. [Online]. Available: https://www.mathworks.com

[20] L. Hazeleger, M. Heertjes, and H. Nijmeijer, "Second-order reset elements for stage control design," in *Proc. Amer. Control Conf.*, 2016, pp. 2643–2648.